Reactive Planning in Air Combat Simulation

Irène Degirmenciyan-Cartault
Dassault Aviation, 78, quai Marcel Dassault Cedex 300 92552 St Cloud Cedex France
irene.degirmenciyan@dassault-aviation.fr

Abstract: In real-world environments such in air combat missions the agents continuously receive perceptual inputs from the environment that is highly dynamic (unpredictable) and uncertain. The aircraft often have to reorganise themselves and to make decisions under time constraints. Two modes of control are thus necessary: planning and reaction. By planning we mean both building a course of action before execution, and reaction as dynamic replanning which interleaves planning and execution. Furthermore, in air combat simulation the plans adopted by the agents in response to external events are known in advance and are not generated by the agents as in other domains. To be reactive, the agents have to choose dynamically the appropriate plans and to coordinate their actions. We present how we take into account new events thanks to dynamic allocation of tasks by means of the graphs of dependencies between agents' activities. Our current work aims to extend this model of tasks and goals by integrating time notions in the selection of action plans and coordinate mechanisms. Operations on plans under time constraints are also examined in order to enable the simultaneous management of several events.

1 Introduction

In real-world environments such as air combat missions the agents continuously receive perceptual inputs from the environment that is highly dynamic (unpredictable) and uncertain. The aircraft often have to reorganise themselves and to take decisions under time constraints. Two modes of control are thus necessary: planning and reaction. By planning we will understand both building a course of action before execution, and reaction as dynamic replanning which interleaves planning and execution. Furthermore, in air combat simulation the plans adopted by the agents in response to external events are known in advance and are not generated by the agents as in other domains. To be reactive, the agents have to choose dynamically the appropriate plans and to coordinate their actions.

The purpose of this paper is to point out the difficulties to coordinate the behaviour of several agents in applications such as air combat simulation. A host of approaches to dealing with reactive planning or multi-agent planning exists, but in the context we are interested, we should simultaneously cope with these two aspects in order to treat the coordination of the agents' activities under time constraints facing to the arrival of new events.

After having outlined the interest of using intelligent agents in the domain of warfare simulations (section 2) and characterised the air combat mission and the requirements in terms of modelling coordination and reactive planning in the distributed context of air combat simulation (section 3). We then focus on reactive and real-time planning (section 4 and 5) and multi-agent planning (section 6). In next sections, we present the work led at Dassault Aviation, and particularly the reactive planning process implemented in the SCALA environment (Cooperative System of Software Autonomous Agent) through simple air missions scenarios. We show how SCALA deals with new events by dynamic allocation of tasks to the agents via the use of graphs of dependencies between their activities, and we also define plan operators enabling a more pre-emptive time management.

2 Thinking with agents

Intelligent software agents have been successfully used for modelling human decision making. In particular, intelligent agents have already shown their suitability for operational analysis of aerial warfare simulators

Paper presented at the RTO AVT Course on "Intelligent Systems for Aeronautics",
held in Rhode-Saint-Genèse, Belgium, 13-17 May 2002, and published in RTO-EN-022.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 JUN 2003	**N/A**	**-**

4. TITLE AND SUBTITLE
Reactive Planning in Air Combat Simulation

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Dassault Aviation, 78, quai Marcel Dassault Cedex 300 92552 St Cloud Cedex France

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM001519. RTO-EN-022, The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	UU	**14**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

[STE 96], or man-in-loop simulation as computer generated forces [ILR 97]. Some studies are going so far as to the concept of interchanging humans and agents [HEI 01], or to simulating human performance while taking into account factors such as experience, attention, workload and stress [LLO 97].
Thus, the use of intelligent agents focused on the modelling of human reasoning. The **BDI** model of rational agency have provided the basic paradigm of much of the research in this domain [RAO 95]. The power of this model is the ability to describe folk-psychological notions of **belief, desire and intention**, which helps to describe some aspects human decision making. Furthermore, the appropriate level of abstraction of this model makes it well understood by the analysts and decision makers who are exactly their users. In particular, intelligent agents have widely proven their utility in the modelling of tactical decision process of pilots and fighter-controllers by easily involving the operational air force personnel in the design and development of these kinds of simulations [HEI 98].

3 Air combat mission characteristics

We assume that aircraft and pilots are modelled by intelligent agents. Modelling an air combat mission implies the following points:

- The agents continuously receive perceptual inputs from the environment that constitutes their beliefs of the world (establishment of the situation awareness).
- The environment is highly dynamic (unpredictable) and uncertain, and the aircraft often have to reorganise themselves (dissolution and formation of new teams). For example, reorganisation is performed when a failure occurs or when an aircraft is destroyed (the others have to reconfigure themselves and reassign their roles).
- The agents have to take decisions under time constraints.
- The plans adopted by the agents in response to external events or to accomplish goals are supplied in advance (generally at the briefing before the mission) and are not generated by the agents as in other domains. The agents have to select an applicable plan in their directory of plans under several conditions (this choice is context dependent).
- The aircraft in teams (sections, divisions, packages) have joint goals to achieve (intercept a bandit), and joint plans (team tactics).
- Each member of a team may respect the constraints imposed by the type of organisational structure, and its role within it. The functional responsibilities adopted by the members depend on the goal assigned to the team. A same aircraft can be leader (organisational role) and escorter (functional role).

In a such context, the problems addressed are the coordination and the synchronisation of agents activities under time constraints. So, we will firstly bring reactive and real-time planning to our attention, and secondly, we will focus on multi-agent planning.

4 Reactive planning

As we saw below, air combat simulation involves coordination, and reactive planning in a multi-agent context. Let us compare conventional planners and reactive planners [ASH 00]:

Conventional planners are characterised by their ability to establish a deterministic sequence of actions for getting from a given initial state to a given goal state. They successfully can be applied to problems that are completely defined in a formal way. The original conventional planners attempt to find a path to reach the goal state; they did not really plan, and can also be considered as search algorithms. They do not exhibit recovery notions and can fail if some uncertainties exist in the initial hypothesis or whether the current state of world changes during the planning process.
By contrast, reactive planners construct plans which are able to respond to a large number of world states while working with some type of monitoring of plan execution that keeps track of the world states at all times to respond accordingly.

The notions of reaction and reactive planning have to be jointly considered. Reactive planning takes place before plan execution, although reaction takes place at execution time. These two types of control have to work together to anticipate and avoid critical situations (pro-active behaviour).

A very important notion in reactive planning is the notion of contingency: "Contingency is any state of the world entered by the executing agent while following a plan; which state should not have occurred as a result of executing the plan up to that point" [ASH 00]. The uncertainty characteristic of real-world domains involves the apparition of contingencies during the plan execution. D.W. Ash and V.G. Dabija classified contingencies in three types related to the limited resources that an agent can use, and according to the action taken at planning time to prepare for their occurrence at execution time:

- Contingencies for which the planner builds complete conditional branches, from the contingency state to the goal state, in the main plan (contingencies with high likelihood of occurrence and requiring elaborate plans to treat them).
- Contingencies for which the agent prepares reactive responses; these may be combined into reactive plans which are integrated in the complete plan (to stabilise the situation in a short time), after what a more extensive planning can be necessary at execution time.
- Contingencies ignored by the agent at planning time, either their treatments can be left for dynamic replanning during execution (contingencies with low likelihood of occurrence, and without short-term disastrous consequences), because they are less important than the below categories and do not require replanning at all, and in last case, when the agent can’t take any action to solve the problem.

It seems obvious that an agent, with limited resources, considered in a real world cannot handle all of the contingencies at planning time (concept of "universal plan"). Fortunately, many of the contingencies can be ignored at planning time (for example those which have a low likelihood of occurrence). The problem for the agent is thus to decide which contingencies require reactive responses and those which can be ignored at planning time.

Two control modes are thus necessary: planning and reaction [HAY 93]. By planning we mean both building a course of action before execution, and dynamic replanning which interleave planning and execution. The most important disadvantage of the planning approach is the inflexibility of the planned behaviour. The agent is constrained to act according to the states of the world strictly specified in the plan and leading it straight to his goal failing at the first not planned disrupting event. The reactive approach is more flexible: the agent acts according to a set of perception-action rules which allow it to respond to a larger set of run-times conditions in short-time. Thus, he does not build a complete solution to the final goal, but can quickly stabilise the situation. Therefore, this model of control provides a less carefully in-depth analysis of the current state and the related action consequences.

These two models complement each other, and have to be implemented together, that is what is called reactive planning. In real-world dynamic environments, where short-term responses are necessary, agents need frameworks for action selection. Such frameworks, detailed in [ASH 00], ensure the choice of the best set of events (contingencies) and reactions to be stored at planning time before execution, according to the perception capabilities (sensors), and the reaction execution mechanisms of the agent. Furthermore, planning and reaction techniques can be used to plan the agent’s sensing activities.

5 Real-time planning

In mission-critical system it is imperative to ensure that the proposed solutions have a correct temporal behaviour before they are deployed, and that these solutions have been elaborated under time constraints. The Maruti operating system [LEV 89] [LEV 90] provides tools to build verifiable real-time system services including hard real-time, distributed operation, and fault-tolerance, which are crucial in mission-critical

systems. In this work, the time-driven approach is used, where the control of the resources is done by the operating system scheduler based on the time elapsed in an operation and on absolute time value. This work has shown that a time-driven design is simpler and more easily verifiable.
This kind of system can support the development of dynamic reaction system, which may guarantee performance characteristics, suitable for mission-critical applications.

An example of such a time-driven system applied to aircraft simulation is the Cooperative Intelligent Real-Time Control Architecture: CIRCA.
CIRCA [MUS 93] is an architecture that aims at monitoring environment changes and agent knowledge modifications in order to pre-empt possible failures in the system. It blends the real-time and reasoning requirements by executing them on two separate components [ASH 00]:

- AIS: the artificial intelligence subsystem, which performs high-level reasoning about tasks and develops low-level control plans.
- RTS: the real-time subsystem, which ensures a predictable behaviour for the guaranteed execution of these (mission-critical) control plans.

A *Control plan* is a cyclic schedule of simple test-action pairs (TAP). A TAP contains temporal data such as worst-case timing data on: how long it takes to test the preconditions (TEST-TIME), and how long to execute actions (ACTION-TIME). In addition, the TAP is associated with a maximum TAP period (assigned during planning), which represents the longest time interval allowed between invocations of the TAP that can still guarantee to avoid failure.

The automatically derived reactive control plan is guaranteed to meet the domain's deadlines and achieve the system's goals. The architecture makes a fundamental distinction between activities with respect to the types of goals they are interested to achieve:

- *Control-level goals*, which are guaranteed to meet domain deadlines, via the predictable execution of the RTS. They are frequently related to system safety, and correspond to hard deadlines derived from physical relationships between agents, and the environment (*i.e.* collision avoidance must be achieved before their deadlines). The priority is always given to those goals.
- *Task-level goals*, which are executed in a less predictable manner, are achieved on a best-effort basis. The system tries to achieve them when possible, but if time pressure or other resource limitations make this impossible, the system is still considered successful. The deadlines are negotiable by the agents.

CIRCA operates by simultaneously planning new control plans in the AIS that cooperates with the TAP scheduler, while the RTS executes existing Control plans. The RTS can also influence the AIS by giving it feedback about changes in the world. In that case, the AIS has to replan a set of actions and a control plan.

Distributed versions of CIRCA have been developed [MUS 98] [KRE 01]. The pre-emption process is distributed and coordinated. In fact, each agent has the possibility to avoid the possible failure of another agent: if an agent A determines that an agent B is tracked by a missile, it will tell agent B to activate its electronic counter-measures to defeat the missile. This means that each agent has special TAPs to take care of event that could arise to others such as the Detection of a missile.

Other works are focused on the process of planning [ATK 96]. Those works aims at reducing the planning time by typing the different states that could occur in the environment. Actually, they try to determine in which case the system has to plan a complete control plan or execute an existing Control plan. The main topic of this work is the time that the system has to react to the events. If the transition to failure comes “fast”, then it executes an existing plan. Otherwise, if the transition is “slow”, it will plan a new Control plan. The time allocated to the planning process dependens on the time before the transition occurs.

CIRCA essentially deals with the safety of the system and not with the achievement of high-level goals. The safety is ensured thanks to a real-time monitoring of the environment, where any changes implies the planning of a new control plan with real-time constraints.

6 Multi-agent planning

An important aspect encountered in Air-Combat simulation is the need for coordinated behaviour modelling. The coordination issue is central in multi-agent systems since the agents that constitute a multi-agent system have to coordinate their activities. A way to enhance agents' cooperation is to give them capabilities that enable to forecast and plan cooperatively their actions. Several works deal with multi-agent planning as a coordination strategy [GEO 83] [DUR 88]. Generally, these works address the post planning. First, the agents generate individual plans and then attempt to merge them in a global plan (the multi-agent plan) where the execution of the local plans is compatible. This post planning process lies on the detection of sub-goals and conflicting situations in order to remove them.

One of the major properties of multi-agent planning is the distribution that distinguishes it from traditional (centralised) planning. Although this notion is inherent to multi-agent systems, it becomes ambiguous when it qualifies planning. Indeed, we have to question on what is really distributed. Distribution may concern either the planning process itself or the resulting plans (or even both) [DUR 99]. So, different types of multi-agent planning can be considered [ELF 01]:

Centralised planning and distributed plans

The planning process is centralised in a specific agent (the coordinator) and the resulting plan is distributed over the agents. The plan is partially ordered and the parallel actions can be concurrently executed by several agents. The coordinator can be designated after a negotiation cycle between the agents. It is the case of the Air Traffic Control application [CAM 83] where the aim of the agents is to build coherent flying plans.
This type of planning makes conflict easy to solve and the convergence to a global solution, but it suffers from the traditional disadvantages of centralised control, such as the lack of robustness or the communication bottleneck.

Besides, this approach is not very reactive. Every occurring problem during the execution of a plan has to be passed to the coordinator who may decide to activate some replanning operations. So, the coordinator should maintain a continuously updated representation of execution states in order to solve the conflicts.

Distributed planning and centralised plan

We assume here that the problem to be solved (*i.e.* the tasks to be achieved) is decomposed in sub-tasks and each of them is planned by an agent. The agents have to interact with each other to synchronise and merge the local plans in order to constitute a unique multi-agent plan.

This approach is also costly in communications and the achievement of a global solution is not guarantied.

Distributed planning and distributed plans

This approach of distributed planning is undoubtedly the most challenging because it does not assume that global plan exists somewhere in the system, and yet the distributed plans are compatible, i.e. their executions do not cause conflicts between agents [DUR 99].

The agents plan their actions and execute them concurrently and autonomously in a shared environment. Events due to the concurrency of actions can occur during execution or plan generation. So, this type of planning requires rich and powerful formalisms that can express parallelism, concurrency, hierarchical goals, etc.

Incremental planning is one of the approaches proposed in the domain. It is assumed that a set of plans is already coherent and that the planning consists in integrating a new plan to be coordinated with others.
In [VMA 92], a taxonomy of the relations between plans is developed and a communication framework allows plan exchanges and negotiations between agents to take into account these relations when a new plan is inserted. This work concerns the simultaneous coordination of two agents.

El Fallah-Seghrouchni and S. Haddad in [ELF 96a] proposes a distributed planning algorithm that simultaneously coordinate several agents. The formal model is based on the partial order of the actions. The plans' coordination consists of adding synchronisation links between actions. The algorithm solves the positive interactions by using the pre and post-conditions, and the negative interactions by synchronising the actions. The model guaranties the feasibility of the multi-agent plan for all total order whatever is the total order extending the partial order. This formalism of plan representation is extended in [ELF 95] [ELF 96b] to enable the algorithm to solve more complex situations. It lies on an extension of the Petri-net formalism to the Recursive Petri-net formalism. It includes the characteristics of recursivity, dynamism, and interleaving of planning and execution. It allows the representation and the reasoning on simultaneous actions and continuous processes (concurrent actions, choice between alternatives, synchronisation, etc.).
This planning model allows the representation of abstract action during the generation of plans. The refinement of this type of action is dynamic (during the execution) and context dependent which allows the dynamic choice of the manner of executing an action. Furthermore, this model guaranties the consistency of the generated plans with the help of efficient algorithms.

A more detailed presentation of multi-agent planning is beyond the scope of this paper. The interested reader could refer to [ELF 01] and [DUR 99].

Let us now present the studies carried out at Dassault Aviation that attempts to introduce reactivity and time constraints in the coordination of the activities of agents. The application considered is the air combat simulation. For the moment, the aim of our work does not go so far as to integrate real-time constraints in the reorganisation or planning process, but we are attempting to introduce time constraints in the choice of plans and in the choice of tasks assignment to agents, and further, in the goals management too, *i.e.* how the goals have to be taken into account to satisfy the temporal objectives fixed in order to successfully accomplish the mission.

In the next sections, we are describing the reactive planning process implemented in the SCALA environment (Cooperative System of Software Autonomous Agent) through air missions' scenarios. Firstly, we are demonstrating how SCALA handles new events by dynamically assigning the necessary tasks that treat them, and secondly we are defining operators on plans enabling a more pre-emptive time constrained management.

7 The reactive planning process in SCALA

7.1 A dynamic assignment of tasks

In SCALA, the designer models the behaviour of the multi-agent system at a high level of abstraction via a graph of dependencies. This graph is constituted by one or more distinct graphs composed of tasks or basic behaviours that have to be accomplished by the agents of the system, and the constraints between them. These are defined by links or connectors expressing notions of synchronism (on the beginning or on the end of several tasks), exclusion (the execution of one task inhibits others'), refinement (several methods can be invoked to execute a same task) and abstraction (a task is composed of others). Another type of constraint is about the number of agents having to perform a given task. At this stage, tasks are not yet allocated to the agents.

The definition of the graph of dependencies provides the knowledge required for managing the cooperation between the agents. The need of collaboration for the execution of some tasks implies dynamic planning during simulation through specified cooperation mechanisms. This on-line process interleaves planning and

execution just like in reactive planning. For example, to fire a missile, the designation of the target can be done by an aircraft, and the fire of the missile by another aircraft. Depending on the available resources, the same aircraft also can accomplish these two actions. This coordination can be either implicit, the agents have joint goals and each of them attempt to accomplish the relevant tasks, or explicit, for example, an agent can ask another agents to help him to accomplish a task because this task requires the synchronisation of their actions.

The assignment of the tasks to the agents is thus made dynamically according to the current situation and the available resources (agents are also assimilated to resources through their skills). So, we give the agents a certain freedom of action that gets more reactivity for the system and allows to avoid some failures such as the use of non-available resources.

7.2 The simulation of an interception mission

In this section, we are describing an interception scenario. Let us assume that an enemy wants to ingress our territory with a "**Bomb target**" goal (fig. 1).
The mission of our aircraft is to protect the territory by intercepting the threats. We assume that a four aircraft division is on alert on a CAP waiting for events such as "**Contact on the radar**". A possible sub-graph associated to this event is modelled in fig. 2. The first task that responds to this event is "**Guided flight**" to achieve the interception. This task has to be executed by at least two agents of the entire division (**$* \geq 2$**). The "Guided Flight" is an interruptible task (grey colour), which implies it can be abandoned as soon as the next task "**Acquisition**" is possible. When the agents have obtained the permission to engage the bandit (pre condition of "Engagement" satisfied), the best-placed agent does it.
SCALA enables the on-line addition of new agents (arrival of a new bandit in the theatre). Let us assume now that the designer creates a new bandit following the same type of graph than the first bandit before this one had been treated. The reception of this new event by the Division implies it has now two bandits to treat. A new goal associated with a new instance of the same graph is generated.

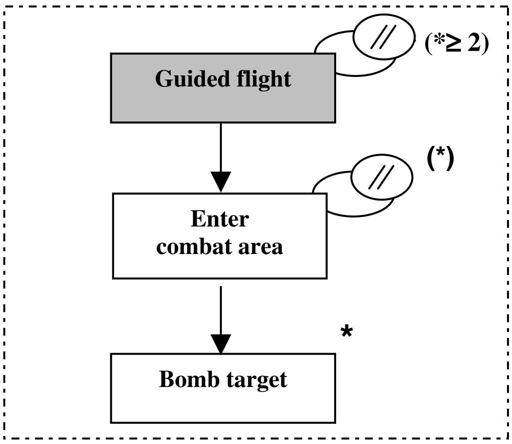


fig. 1: Sub-graph "Bomb Target"

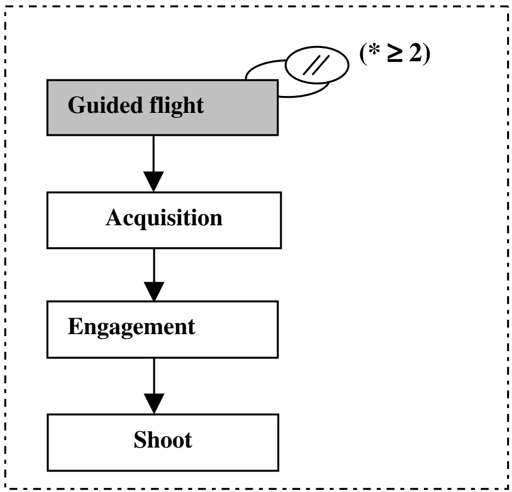


fig. 2: Interception Sub-Graph

In our domain, plans are predefined, so we do not have to generate new plans to respond simultaneously to several events, but only to manage their combination. In the next section, some of our operators on plans will be defined.

We favour that goals must be treated concurrently if the agents have enough resources. The agents are considered as resources (skills and role), and physical resources of the agents, such as missiles are also taken into account. In the other case, the priority of urgency of the goals has to be deal with. The complete heuristic of goals' management is explained beside (fig. 3).

In our scenario, we assume the division has enough resources to treat simultaneously the two threats. The division thus split into two groups: Section1 assigned to the first target, Section2 assigned to the second target. This decision is made by the leader of the division according to the structural organisation specified by the designer.
The difficulty here is to re-organise the aircraft in the best way to take into account the new goal. This reorganisation depends on many parameters (context, physical and temporal resources). The problem is to extract the best required criteria to ensure that each new group (reconfiguration inside the groups is possible) has the minimal resources to achieve its assigned goal. This re-organisation is decisive for the continuation of the mission and requires a time-constrained pre-planning algorithm on a high-level of decision.

```
Priority(goal G1, goal G2)
Begin
        If enough Resources
                Treat G2 and G1 concurrently;
        End If
        If  not enough Resources
                If P2 > P1
                        Treat G2 then G1;
                End If
                If P2 ≤ P1
                        Treat G1 then G2;
                End If
        End If
End
```

fig. 3: Goals management algorithm

Then, Bandit2 ripostes (he replies also to an event such as “Contact on the radar” by a sub-graph “Counterattack”) and we assume to simplify the scenario that he is immediately shot by Section2. This Section examines if there is some bandits left. That is why they consider Bandit1 as their new target and decide to follow him. The two sections have now the same goal. Finally, Section1 shoots Bandit1, and they can return home because there are no more bandits in the friend territory. The mission is completed.

This scenario is depicted in fig. 4, 5 and 6 on a visualisation tool enabling to situate the agents and show their behaviour. SCALA provides other tools, independent of the application, to analyse the behaviour of the system (dynamic information on the states of the agents and on their activities).

SCALA has been developed on top of the JACK agent-oriented language [HOW 01], itself built on top of Java.

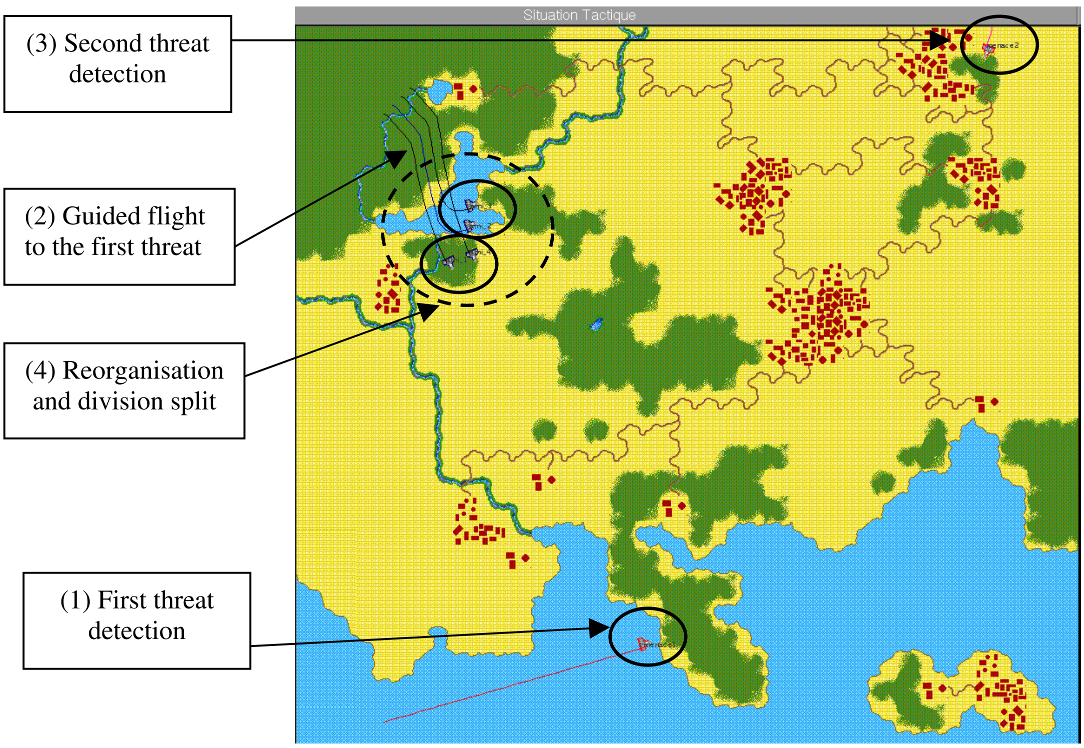


fig. 4: An air combat scenario developed in SCALA (1)

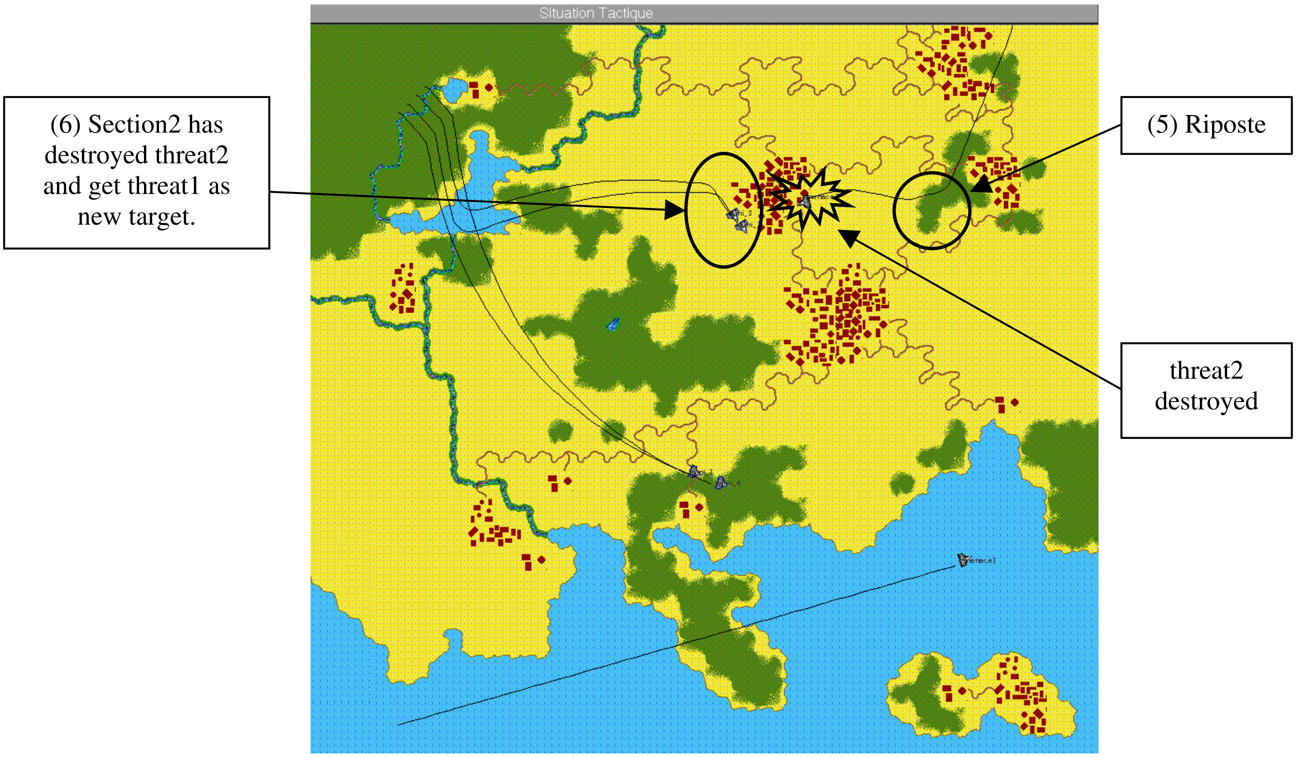


fig. 5: An air combat scenario developed in SCALA (2)

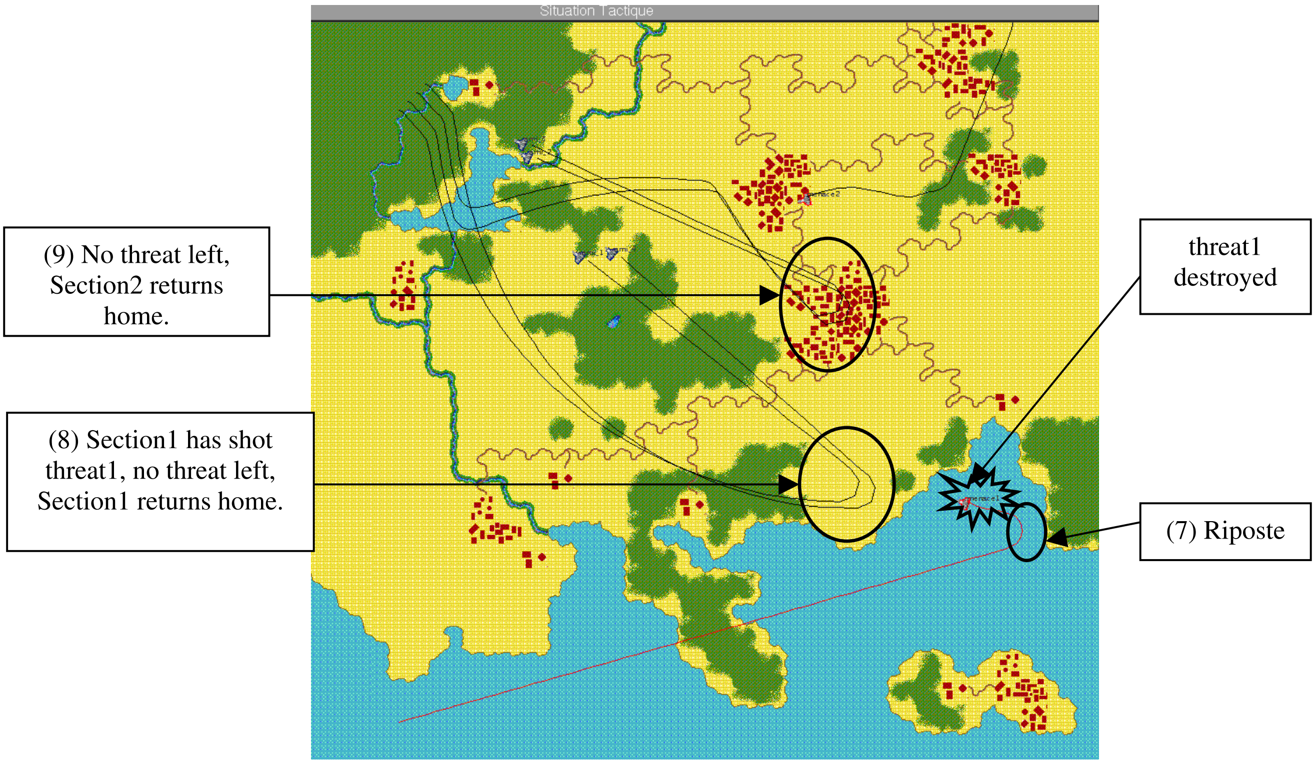


fig. 6: An air combat scenario developed in SCALA (3)

Several works have been developed in multi-agent simulations like SWARMM [HEI 98] [MIL 96] based on dMARS [DIN 97] or TacAir-Soar [ROS 94].

SWARMM is a detailed multi-agent simulation of fighter combat designed for analysing the impact of upgrades to modifications and development of the tactical employment of aircraft. It is capable of simulating the physics of air missions and the pilot's reasoning process involved in such missions. In SWARMM, the behaviour of the agents is based on the choice of relevant pre-specified plans supported by the meta-level reasoning of the agent-oriented software dMARS. So, the process of replanning is entirely based on the direct choice of behaviours through predefined plans, *i.e.* all the behaviours, even team tactics are implemented in specific plans [TID 98]. The essential difference between the graph of dependencies of SCALA and SWARMM plans, is that the choice of behaviours in SCALA is less determined, or more exactly let at the last moment. The graphs are a kind of factorisation of plans, and represent a class of behaviours responding to a same goal. It is the coordination mechanisms that instantiate at the last moment the graph, which becomes a plan. This approach allows making lighter the definition of the plans and their pre-conditions. In SCALA, a first level choice is made for the relevant graph, and a second level choice is made by the agents themselves through the heuristics of coordination.

The TacAir-Soar system also combines reactive and goal-driven reasoning. It contains a large set of rules that fire as soon as their conditions are met, without search or conflict resolution. The choice of the actions to be performed entirely lies on the interpretation of the environment observations [JON 94]. Another part of this research effort deals with a deliberative planning component that separate planning from normal execution by projecting future possible states and searching through them which courses of action are appropriate. The challenge of this research is to integrate deliberative planning with dynamic reasoning.

7.3 Time constraints

Our current work in collaboration with the LIPN (Laboratory of Computer Science of Paris 13) consists to extend the SCALA framework with temporal aspects in order to satisfy time constraints at the execution level. Let us now introducing the expected improvements of SCALA.

A part of our current research effort is to equip the graph of dependencies with the notion of time. The graph or more precisely the sub-graphs can be constrained in terms of temporal objectives. In fact, the global time constraint on a graph is the resultant of time constraints on its tasks. The main idea is that time constraints can be added to the graph, giving a limited duration of the plan execution. This notion is crucial in the context of a real-time environment where the execution time allocated to agents to accomplish tasks is critical. Our approach is real-time execution driven.

Time constraints are defined as intervals of time to be associated with the tasks as in [VMA 92]. We place our work in the framework of real-time execution that is to say that the agents have temporal objectives on the execution of their tasks. We attempt to develop an anytime approach for task execution such as the "contract algorithms" [ZIL 96]. We think that temporal objectives should be an important criteria in the process of planning because the behaviour of a group of agents depends on the time it has to react. So, the available time must be considered as a resource and the choice of the relevant graph must take into account both physical and temporal resources.

7.4 Operations on plans

The choice of the relevant operations is time dependant. We take the hypothesis that we will always try to treat all the goals concurrently if there are enough resources in the system. The following section is devoted to describe the operators on plans and finally depicting several behaviours.
Before choosing what type of operation an agent is allowed to do, he verifies if he can maintain the global goal. In fact, an agent can have different behaviours: to keep his current plan and to cope with the new allocated tasks or to distribute his current plan (i.e. set of tasks) over the other agents. In the last case, we consider that the agent does not maintain the global goal himself but this goal is maintained by the others. In

an extreme case, the agent cannot maintain the global goal, *i.e.* he cannot keep his initial set of tasks and simultaneously manage the new tasks.

Global goal is maintained

We have developed several operators to manipulate the plans: concatenation, parallelisation, and insertion (merging). These operators are invoked when the global goal is maintained:

- *Concatenation*: This operator sequences two plans. The interest of a simple concatenation is limited. In fact, it would be interested, in a general manner, to re-examine the two plans, by adding or deleting tasks such as redundant tasks. With such an operator, the dependencies are not updated.
- *Parallelisation*: The two plans are performed concurrently. To use this operator, possible conflicts about resources have to be checked. In this case, the dependencies are not updated because they do not interfere with the global goal. So, no updating with other agents is necessary.
- *Insertion*: The insertion of plan consists in adding a new set of tasks into the current plan. This implies an updating of the dependencies with the other agents. In fact, even if the global goal is not changed in terms of objectives, the constraints on those objectives (such as the delays of execution) could be modified.

Global goal is not maintained

When the global goal is not maintained, the agent has two possible behaviours. Actually, its behaviour depends if it evolves alone or with others. In the first case, if it cannot cope with the new goal, also it removes it and starts again from a *home state*. But in the second case, it distributes its current set of tasks over the other agents of its group.

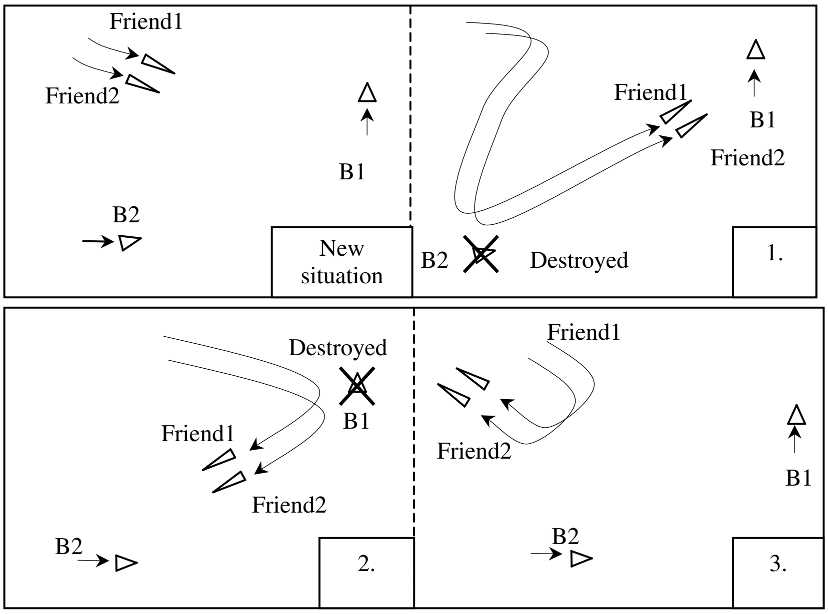
7.5 Some examples of behaviours

The presented scenario involves two "Friend" planes (Friend1 and Friend2) composing a patrol and firstly, a single bandit B1. The threat is following a flight plan and the patrol has a goal that is "To intercept the threat". As in the first scenario, the patrol fulfils a classic plan to intercept this menace. The patrol succeeded when the threat is destroyed. In this scenario, we create a new bandit B2 following a flight plan too. This involves the arrival of a new goal for the patrol that takes it into account.
Several alternatives should be applicable to cope with the arrival of a new event (fig. 7):

1. To keep the entire patrol and to insert the goal if its priority is higher than the current one (the insertion operator is used)
2. To keep the entire patrol and to concatenate the goal if its priority is lower or equivalent than the current one (the concatenation operator is used)
3. To keep entire the patrol and to abandon the goal an to start again from a home state if there is no available resources or if the friends are outnumbered (the goal is not maintained)
4. To split the patrol and to simultaneously treat the two bandits if there are enough available resources to treat them concurrently (the parallelism operator is used).

Actually, it is obvious that treating the two goals concurrently takes less time than sequentially. But this point could also be tactical since a patrol is generally stronger than an isolated aircraft. As a consequence, if there is enough available time for the patrol to achieve sequentially the goals, it could sometimes be better than splitting it. We assume that those operational aspects are under the responsibility of the designer and he has to specify them in the graph of dependencies.

Work in progress tackles the issue of validation of the generated plans after the operations. Our approach is based on timed automaton. They present the advantages to model some of our connectors as the refinement and the abstraction [ALU 94] [BER 97] based on different semantics of time, considering the action as instantaneous or with duration. In this approach, the multi-agent plan is modelled as a synchronised network of timed automata where each agent plan as a timed automaton. The main interest is to control plans execution while taking into account the action duration and the arrival of new events.



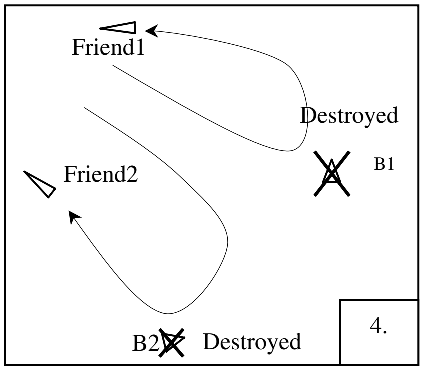


fig. 7: Multi-agent behaviours

8 Conclusion

In real-world environments such as air combat mission the agents continuously receive perceptual inputs from the environment that is highly dynamic (unpredictable) and uncertain. The aircraft often have to reorganise themselves and to take decisions under time constraints. In this paper we focused on the multi-agent technology that provides an interesting framework to design human-like behaviours and to model collective behaviours and also on reactive planning that is appropriated to deal with the related domain which requires both reactive and pro-active behaviours.

The SCALA project based on the multi-agent paradigm proposes a functional approach to design complex systems and provides tools to rapidly setting up simulations. The designer is supported by tools enabling the modelling of high-level behaviours through the graphs of dependencies that contain the necessary information to coordinate agents' activities. Indeed in air combat simulations the plans adopted by the agents in response to external events are known *a priori* and are not generated by the agents as in other domains, but the agents have to dynamically choose the relevant plans and then to coordinate their actions.

Our current work led in collaboration with the LIPN (Laboratory of Computer Science of PARIS 13) focuses on the extension of the SCALA graphs to temporal aspects, that have to be taken into account in the agents' plans and in the coordination mechanisms. Indeed the agents in such applications have to respect duration on their tasks or on the achievement of their joint goals. Operations on plans under time constraints are also examined to enable the simultaneous management of events. These operators enable the agents to cooperate through the concatenation, the insertion or the parallelisation of their plans. The formalism that we are exploring is close to the timed automata and should allow to easily model some of our connectors such as the refinement and abstraction connectors and to validate the generated plans [ALU 94] [BER 97].

Acknowledgements

The author would like to thank Amal EL FALLAH-SEGHROUCHNI, researcher at the LIPN (Laboratory of Computer Science of PARIS 13) for her collaboration on this work with Dassault Aviation and discussion. She also would like Frederic MARC who works on this project in the framework of his Ph.D. on Computer Science at LIPN and Dassault Aviation.

Reference

[ALU 94] ALUR R., DILL D. L., "A Theory of Timed Automata", *Theoretical Computer Science* Vol. 126, p. 183-235, 1994.

[ASH 00] ASH D. W., DABIJA V. G., ."Planning for Real-Time Event Response Management", Edited by Prentice Hall PTR, 2000.

[ATK 96] ATKINS E. M., DURFEE E. H., SHIN K. G., "Detecting and Reacting to Unplanned-for World States", 1997.

[BER 97] BERARD B., GASTIN P. and PETIT A., "Refinement and abstraction for timed languages", Research Report LSV-97-3, Apr. 1997.

[CAM 83] CAMMARATA S., McARTHUR D., STEEB R., "Strategies of Cooperation in Distributed Problem Solving", In *Proceeding of the 8th International Joint Conference On Artificial Intelligence,* pp. 767-770, Karlsruhe, Deutschland, 1983.

[DIN 97] D'INVERNO M., KINNY D., LUCK M., WOOLDRIDGE M., "A formal specification of dMars", In M. Wooldridge and A. Rao, editors, Fourth International *Workshop on Theories, architectures and Languages*, 1997.

[DUR 88] DURFEE E. D., ."Coordination of Distributed Problem Solvers", Kluwer Academic Press, Boston 1988.

[DUR 99] DURFEE E. D., ."Distributed Planning ", In *Multi-agent Systems: A modern Approach to Distributed Artificial Intelligence*, MIT Press, pp. 139-149, 1999.

[ELF 95] EL FALLAH-SEGHROUCHNI A., HADDAD S., "A Formal Model for Coordinating Plans in Multi-Agents Systems", In *Proceeding of Intelligent Agents Workshop,* Augusta Technology Ltd, Brooks University, Oxford, Grande-Bretagne, 1995.

[ELF 96a]. EL FALLAH-SEGHROUCHNI A., HADDAD S., "A Coordination Algorithme for Multi-Agents Planning", In *Proceeding of MAAMAW'96,* Springer-Verlag, LNAI 1038, Eindhoven, Pays-Bas, 1996

[ELF 96b]. EL FALLAH-SEGHROUCHNI A., HADDAD S., "A Recursive Model for Distributed Planning", In *Proceeding of ICMAS'96 (International Conference of Multi-Agnets Systems),* AAAI Press, Kyoto, Japan, 1996

[ELF 01] EL FALLAH-SEGHROUCHNI A., "Modèles de coordination d'agents cognitifs", In *Principes et architectures des systèmes multi-agent*, Edited by Hermes, pp. 139-176, 2001.

[GEO 83] GEORGEFF M. P., ."Communication and Interaction in Multi-Agents s of ", In *Proceeding of the Third National Conference on Artificial Intelligence (AAAI-83)*, pp. 125-129, July 83.

[HAY 93] HAYES-ROTH B., "Opportunistic Control of Action in Intelligent Agents", *IEEE Transactions on Systems, Man and Cybernetics*, Special Issues on Planning, Scheduling, and Control: An AI-based Integrated System View, March 1993.

[HEI 98] HEINZE C., SMITH B., CROSS M., "Thinking Quickly: Agents for Modelling Air Warfare", In Proceedings of Australian Joint conference on Artificial Intelligence, AJCAI 98, Brisbane, Australia, 1998.

[HEI 01] HEINZE C., GOSS S., JOSEFSSON T., BENNETT K., WAUGH S., LLOYD I., MURRAY G., OLDFIELD J., "Interchanging Agents and Humans in Military Simulation", In AAAI Workshop on the Innovative Application on AI, 2001

[HOW 01] HOWDEN N., RONNQUIST R., HODGSON A. and LUCAS A., "JACK Intelligent Agents™ Summary of an Agent Infrastructure", In *Proceedings of the fifth Conference on Autonomous Agent*, 2001.

[ILR 97] McILROY D., HEINZE C., APPLA D., BUSETTA P, TIDHAR G., RAO A., "Towards Credible Computer-Generated Forces", In *Proceedings of The International Simulation Technology and Training conference*, (SimTecT 97), 1997.

[JON 94] JONES R. M., LAIRD J. E, "Multiple information sources and multiple participants: Managing situational awareness in an autonomous agent", In *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*, May 1994.

[KRE 01] KREBSBASCH K. D., MUSLINER D. J., "Multi-Agent Mission Coordination via Negotiation*", Working Notes of the AAAI Fall Symposium on Negotiation Methods for Autonomous Cooperative Systems*, 2001.

[LEV 89] LEVI S. T., TRIPATHI S. K., CARSON S. D., AGRAWALA A. K., "The MARUTI Hard Real-Time Operating System", ACM Operating System Review, Vol. 23, N°3, June 1989.

[LEV 90] LEVI S. T., AGRAWALA A. K., "Real-Time System Design", McGraw Hill Publishing Co., New-York, 1990.

[LLO 97] LLOYD I., "Simulating Human Characteristics for Operational Studies", DSTO Research Report DSTO-RR-0098, 1997.

[MIL 96] McILROY D., HEINZE C., "Air Combat Tactics in the Smart Whole AiR Mission Model", In *Proceedings of the First International Simulation Technology and Training conference*, (SimTecT 96), Melbourne, Australia, 1996.

[MUS 93] MUSLINER D. J., "CIRCA: The Cooperative Intelligent Real-Time Control Architecture" *Ph. D. Dissertation*, University of Michigan, Ann Arbor. Available as University of Maryland Computer Science Technical Report CS-TR-3157.

[MUS 98] Issues in Distributed Planning for Real-Time Control D. J. Musliner, K. D. Krebsbasch, M. J. S. Pelican, R. P. Goldman, M. S. Boddy, In *Working Notes of the AAAI Fall Symposium on Distributed Continual Planning*, Oct. 1998

[RAO 95] RAO A. S., GEORGEFF M. P., "Modeling rational agents within a BDI architecture", In J. Allen, R. Fikes, and E. Sandewall, editors, In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, pp. 473-484. Morgan Kaufman Publishers, San Mateo, 1991.

[ROS 94] ROSENBLOOM P. S., JOHNSON W. L., JONES R. M., KOSS F., LAIRD J. E, LEHMAN J. F., RUBINOFF R., SCHAMB K. B., TAMBE M., "Intelligent automated agent for tactical air simulation: A progress report", In *Proceedings of the fourth Conference on Computer Generated Forces and Behavioral Representation*, May 1994

[STE 96] STEUART S., MURRAY G., TIDHAR G., RAO A., "Air Mission Modelling - An Application for Articicial Intelligence", In *Proceedings of The First International Simulation Technology and Training conference*, (SimTecT 96), Melbourne, Australia, 1996.

[TID 98] TIDHAR G., HEINZE C., SELVESTREL M., "Flying Together: Modelling Air Mission Teams", *Applied intelligence*, vol. 8, pp. 195-218, 1998.

[VMA 92] VON MARTIAL F., "Coordinating Plans of Autonomous Agents", Springer-Verlag, LNAI, 1992.

[ZIL 96] ZILBERSTEIN S., "Using Anytime Algorithms in Intelligent systems", In *Proceedings of the National Conference on Artificial Intelligence (AAAI),* 1996.